IN THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application. Claims 13 and 21 have been amended.

1. (withdrawn) A method of adding an element to a data structure, comprising:

atomically associating the element with the data structure if the element is not associated with the data structure;

preparing the element for insertion into a location in the data structure; and

atomically inserting the element into the location in the data structure if another element has not been inserted into the location.

2. (withdrawn) The method of claim 1, further comprising:

determining whether the element is associated with the data structure.

3. (withdrawn) The method of claim 2, wherein the atomically associating is performed after the determining and includes:

ascertaining whether the element is associated with the data structure, and

associating the element with the data structure if the ascertaining determines that the element is not associated with the data structure.

4. (withdrawn) The method of claim 1, wherein the preparing includes:

pointing the element toward the location in the data structure.

5. (withdrawn) The method of claim 2, wherein the atomically inserting includes:

determining whether another element has been inserted into the location in the data structure, and

inserting the element into the location by pointing the data structure toward the element the determining determines that another element has not been inserted into the location.

6. (withdrawn) The method of claim 1, further comprising:

repeating the preparing the element for insertion and the atomically inserting the element if another element has been inserted into the location.

7. (withdrawn) The method of claim 1, further comprising:

continuing to add elements to the data structure if an executing thread among two or more executing threads is suspended or killed at any point in the method.

8. (withdrawn) The method of claim 1 wherein the atomically associating, the preparing, and the atomically inserting are based on a field in the element that points to a next element in the data structure.

9. (withdrawn) An article of manufacture, comprising:

a storage medium having instructions stored thereon that, when executed by a computing platform, may result in addition of an element to a list by:

checking whether the element already exists in the list;

performing an uninterruptible operation to associate the element with the list if the checking determines that the element does not already exist in the list, the uninterruptible operation including:

determining whether the element is already associated with the list, and

associating the element with the list if the determining determines that the element is not already associated with the list; and

inserting the element into the list.

10. (withdrawn) The article of manufacture of claim 9, wherein the instructions, when executed, further result in not adding the element to the list when the checking determines that the element already exists in the list or when the determining determines that the element is already associated with the list.

11. (withdrawn) The article of manufacture of claim 9, wherein the instructions, when executed, result in the inserting the element into the list by:

pointing the element to a target element in the list, and

performing another uninterruptible operation to insert the element into the list if the target element in the list has not changed.

12. (withdrawn) The article of manufacture of claim 11, wherein the another uninterruptible operation includes:

determining whether the target element in the list has changed, and

pointing the list to the element if the target element in the list has not changed.

13. (currently amended) A tangible machine readable medium having stored thereon data which when accessed by a machine causes the machine to perform a method, the method comprising:

during concurrent execution of a first and second thread, determining via the first thread whether an existing data element is not a member of a data structure; and

performing an atomic store operation to add the data element to the data structure via the first thread only if the data element is not already a member of the data structure, otherwise performing a next processing task via the first thread;

wherein said determining further comprises determining whether a pointer field value of the existing data element is null.

14. (previously presented) The machine readable medium of claim 13, the method further comprising: performing a second atomic store operation, responsive to determining that the pointer field value is null, to set the value of the pointer field value of the element to a non-null value.

15. (previously presented) The machine readable medium of claim 13, the method further comprising:

during concurrent execution of the first and second thread, determining via the second thread whether the element is a member of the data structure;

wherein said determining further comprises determining whether the pointer field of the element is null.

16. (previously presented) The machine readable medium of claim 15, the method further comprising:

executing a subsequent processing task via the second thread, responsive to determining that the pointer field of the element is not null;

wherein the subsequent processing task performs processing other than adding the element to the data structure.

17. (previously presented) The machine readable medium of claim 13, wherein the atomic store operation is a compare and exchange operation.

18. (previously presented) The machine readable medium of claim 15, the method further comprising:

performing an atomic store operation to add a second element to the data structure via the second thread, responsive to determining that a pointer field value of the second element is null.

19. (previously amended) The machine readable medium of claim 13, the method further comprising:

initializing the pointer field value of the element to a null value.

20. (previously presented) The machine readable medium of claim 13, wherein the data stored on the machine readable medium comprises compiled program instructions.

21. (Currently amended) A method, comprising:

executing one or more program instructions that control program flow for an executing thread during concurrent execution of the thread and one or more other threads;

wherein the one or more program instructions cause a branch to be taken in response to determining that a pointer value for an existing data element is not equal to an initialization value;

wherein the branch further comprises a branch around one or more atomic operations to add the existing data element to the data structure, such that the atomic operations are not executed; and

wherein the one or more program instructions otherwise cause the atomic operations to be executed responsive to determining that the pointer value is equal to the initialization value.

22. (previously presented) The method of claim 21, further comprising:

executing the one or more program instructions during execution of one or more of the other threads, such that the one or more of the other threads are prevented from adding the element to the data structure multiple times during execution.

23. (previously presented) The method of claim 21, wherein the data structure is a linked list, and

wherein the atomic operations are implemented using the pointers value.

24. (previously presented) The method of claim 21, wherein the one or more atomic operations further include:

determining whether the pointer value for the element is still equal to the initialization value, and

assigning a different value to the pointer.

25. (previously presented) The method of claim 24, wherein the different value is a value that points to the element itself.

26. (previously presented) The method of claim 21, further comprising:

executing the one or more program instructions during execution of one of the other threads, such that the other thread adds a second element to the data structure during execution of the one or more atomic operations.

27. (previously presented) The method of claim 23, wherein the one or more atomic operations further include:

determining whether a head pointer of the linked list has a current value equal to the pointer value for the element, and

assigning to the head pointer a value that points to the element.